Dynamic Database Reordering System

CROSS REFERENCE TO RELATED APPLICATIONS

This application claims benefit of U.S. Provisional Patent Application Ser. No. 60/397,253, filed on 18 July 2002.

BACKGROUND OF THE INVENTION

TECHNICAL FIELD

The invention relates to the ordering of elements extracted from a database. More particularly, the invention relates to the ordering of displayed elements from a database through the ranking of database elements that are actually selected by a user.

DESCRIPTION OF THE PRIOR ART

Technological advances have enabled manufacturers to create various small-format personal electronic devices. Some examples are Personal Data Assistants (PDA), Cellular Phones, small-form-factor data entry units, and other small-form-factor communication units.

As the size of these small electronic data devices decreased, the size of the data entry keyboards on the devices shrank. The solution to reducing the keyboard size was to decrease the number of keys on the keyboard. Reducing the number of keys has created several problems. The most obvious is the overloading of keys such as on a cellular phone. A single key may represent several characters. When text is input into a reduced keyboard device, it becomes tedious and difficult for the user to enter any reasonable amount of text. The overloaded keys typically require multiple presses to obtain the correct characters.

Keyboard disambiguating systems such as described in U.S. Patent Nos. 5,818,437, 5,953,541, 6,011,554, and 6,286,064 owned by the Applicant solve the text entry problem by processing user keystrokes and forming and presenting words to the user that are associated with the keys pressed. Complete words are presented to the user

that begin with the letters represented by the key presses. Presenting a list of words associated with the keys pressed saves the user from entering additional keystrokes to spell an entire word and also saves time. The user simply selects the first word in the list or scrolls down and selects the desired word.

The words that are presented to the user are stored in a vocabulary database. An example of a vocabulary database is described in U.S. Patent Nos. 5,818,437, 5,953,541, 6,011,554, and 6,286,064 owned by the Applicant.

Another example is iTap by Motorola, Inc. of Schaumburg, IL, which performs predictive keypad text entry on cellular phones. The iTap system also displays predicted words to the user. However, iTap does not order the displayed words to the user based on which words were actually used by the user. Such a feature would be extremely helpful to the user to save even more time and enable the user to enter text more quickly and efficiently.

It would be advantageous to provide a dynamic database reordering system that displays words associated with key presses to a user in an order based on the user's actual use of the words. It would further be advantageous to provide a dynamic database reordering system that does not store frequency of use information in the main database.

SUMMARY OF THE INVENTION

The invention provides a dynamic database reordering system. The invention displays words associated with key presses to a user in an order based on the user's actual use, if any, of the words. In addition, the invention does not store frequency of use information in the main database, thereby requiring minimal storage space.

A preferred embodiment of the invention provides a linguistics database that contains words that are ordered according to a linguistics model that dictates the order in which words are presented to a user. A user enters keystrokes on a keypad of a communications device. While the user is pressing keys, the invention predicts the words, letters, numbers, or word stubs that the user is trying to enter. Complete words are dynamically displayed to the user that begin with the letters represented by the key presses. The user typically presses a sequence of keys which is associated with more that one word in the database. In order to save space storing the linguistics database, the linguistics database is pre-ordered before placement into the product.

The invention provides for reordering of the linguistics model order based on the user's usage of the system. If more than one word shares the same key sequence, the most commonly used word is presented as a first choice in the displayed list. If the user does not want that word, but wants another word that is associated with the key sequence, then the user has the ability to scroll through the displayed list of words by pressing a next key, or scroll up or down keys. Once the user has found the desired word, the user activates a select key and the system enters the desired word into the user's text message at the insertion point.

A preferred embodiment of the invention tracks the user's word selections. Once a word has been selected as a result of a next key selection (the nexted word), a frequency value is applied to the selected word and the word ordered first by the linguistics model in the linguistics database for that key sequence.

The first time that a word is nexted by the user, the frequency value of the nexted word is typically lower than the frequency value of the first ordered word in the displayed list. The next time the nexted word is nexted to again, the frequency value of the nexted word is increased relative to the frequency value of the first ordered word. The frequency values are adjusted every time a word is selected.

Another embodiment of the invention looks at the position of the nexted word in the displayed list. If the nexted word is positioned after the second word in the displayed list, then the nexted word is promoted to the second word position. This increases the nexted word's frequency to the second word's frequency or a frequency above the second word's frequency, but below the first word's frequency.

The frequency value of the nexted word will become greater than the frequency value of the first displayed word upon nexting to the same word over and over again. Subsequent user entries of the key sequence for the nexted word and the first ordered word will result in displaying the nexted word before the word ordered first by the linguistics model.

In one embodiment of the invention, a word's frequency becomes greater than the first ordered word or another word with a higher frequency, when that word has been nexted to three (or a predetermined number) more times than the first ordered word.

Other aspects and advantages of the invention will become apparent from the following detailed description in combination with the accompanying drawings, illustrating, by way of example, the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a diagram of a portable communication device with a reduced keyboard according to the invention;

Fig. 2 is a diagram of a cellular phone keyboard according to the invention;

Fig. 3 is a block schematic diagram of an task viewpoint of the invention according to the invention;

Fig. 4 is a block schematic diagram of a linguistic, manufacturer, and user database components according to the invention; and

Fig. 5 is a schematic diagram of frequency information stored in a user database according to the invention.

DETAILED DESCRIPTION OF THE INVENTION

The invention is embodied in a dynamic database reordering system. A system according to the invention displays words associated with key presses to a user in an order based on the user's actual use, if any, of the words. The invention additionally does not store frequency of use information in the main database, thereby requiring minimal storage space.

A preferred embodiment of the invention provides a method for displaying results retrieved from a linguistic database to a user that match the user's key presses on a keyboard. Any word selected by the user from the displayed results is assigned a frequency value that is determined by the user's word usage patterns. The frequency value is preferably not stored in the main linguistic database and only words that are actually used by the user are assigned a frequency. Some words that are used by the user that do not have the possibility of collisions with other words do not need to have a frequency assigned. The method reduces the amount of memory required to 1/7 of what would typically be needed to track a user's usage.

Referring to Fig. 1, a personal communications device 101 with a reduced keyboard is shown. Keyboard disambiguating systems such as described in U.S. Patent Nos.

5,818,437, 5,953,541, 6,011,554, and 6,286,064 owned by the Applicant solve the text entry problem where input keys 102 are overloaded and a single key may represent several characters. User keystrokes on the keypad 102 are processed and displayed 103. While the user is pressing keys, the system predicts the words 106, 107, letters 108, numbers 109, or word stubs (not shown) that the user is trying to enter. Complete words are dynamically displayed 105 to the user that begin with the letters represented by the key presses.

Presenting a list of words associated with the keys pressed saves the user from entering additional keystrokes to spell an entire word and also saves time. The user selects the first word in the list or scrolls down the list and selects the desired word. In this example, the user can press the space key 112 to accept the first word in the list. The selected word appears in the user's text entry position 104. The user can also continue to press keys to further narrow or refine the selection of words displayed.

The words that are presented to the user are stored in a vocabulary database. If the words in the database are sorted by the frequency of use of each word, then the same words are always presented 105 to the user in the same order.

The invention's T9® linguistics database (LDB) contains words that are ordered according to a linguistics model that dictates the order in which words are presented to a user. The user typically presses a sequence of keys which is associated with more that one word in the database.

In order to save space storing the LDB, the LDB is pre-ordered before placement into the product. The words are ordered using a linguistics model that measures the commonality frequency value for each word in the database. The database is assembled using the frequency ordering. The frequency values are not stored with the words in the database once it is compiled, thereby requiring less space to store the LDB.

For example, Fig. 2 shows a typical cellular phone keyboard 201. If the linguistics model indicates that the word "in" has a higher frequency that "go", then "in" appears before "go" in the display list when the 4 key 202 and 6 key 203 on a conventional cellular phone keyboard are selected in that order.

The invention provides for reordering of the linguistics model order based on the user's usage of the system.

Referring again to Fig. 1, if more than one word shares the same key sequence, the most commonly used word is presented as a first choice in the displayed list. If the user does not want that word, but wants another word that is associated with the key sequence, then the user has the ability to scroll through the displayed list of words 105 by pressing the 0 or NEXT key, or scroll up or down keys, if present. Once the user has found the desired word, the user activates a select key, or space key 112, and the system enters the desired word into the user's text message 103 at the insertion point 104.

A preferred embodiment of the invention tracks the user's word selections. Once a word has been selected as a result of a NEXT key selection, a frequency value is applied to the selected word and the word ordered first by the linguistics model in the LDB for that key sequence. The frequency values applied to the word ordered first by the linguistics model and the word that has been selected as a result of a NEXT key (the nexted word) are dependent upon a number of factors. An example factor includes the commonality of usage of the nexted word relative to the first ordered word.

The first time that a word is nexted by the user, the frequency value of the nexted word is typically lower than the frequency value of the first ordered word in the displayed list. The next time the nexted word is nexted to again, the frequency value of the nexted word is increased relative to the frequency value of the first ordered word. The frequency values are adjusted every time a word is selected.

Another embodiment of the invention looks at the position of the nexted word in the displayed list. If the nexted word is positioned after the second word in the displayed list, then the nexted word is promoted to the second word position. This increases the nexted word's frequency to the second word's frequency or a frequency above the second word's frequency, but below the first word's frequency. Collisions will be discussed later in the text.

At some point upon nexting to the same word over and over again, the frequency value of the nexted word will become greater than the frequency value of the first displayed word. Thus, subsequent user entries of the key sequence for the nexted word and the first ordered word will result in displaying the nexted word before the word ordered first by the linguistics model.

In one embodiment of the invention, a word's frequency becomes greater than the first ordered word or another word with a higher frequency, when that word has been selected to three (or a predetermined number) more times than the first ordered word.

In another embodiment of the invention, words and their order are stored in a linguistics database and the frequency values determined as a result of nexting are stored in a separate database (a reorder database). The order identified by the frequency values in the reorder database takes priority over the order of words in the linguistics database. When frequency values are stored in the linguistic database, the frequencies in the linguistic database provide initial values for the frequencies in the reorder database.

In yet another embodiment of the invention, if no frequency values are stored in the linguistic database, the order of the words in the linguistic database are used to synthesize initial values for the frequencies in the reorder database.

With respect to Fig. 3, a subsection of the invention's text processor is shown. The Keyboard Manager 303 monitors the user's key presses. Each key press is sent to the Database Manager 301. The Database Manager 301 gathers each key press and performs predictive word processing.

The Database Manager 301 accesses the linguistic database on the host device's storage device 304 and forms a predictive word list by extracting the first n number of words form the linguistic database that match the keys pressed. The value n is dependent upon the length of list preferred by the manufacturer or user. If the list is too long, then the number of keystrokes that it takes to scroll through the list would be much greater than the total amount of key presses required to type in a complete word. The Database Manager 301 passes the list of words to the Display Manager 302. The Display Manager 302 displays the list of words to the user.

Each time a scroll key or select is pressed, the Keyboard Manager 303 notifies the Display Manager 302. The Display Manager 302 highlights the proper word in the display list using the scroll key presses. When the user presses a select key, the Display Manager 302 inserts the selected word into the user's text entry field and notifies the Database Manager 301 which word was selected.

The Database Manager 301 adjusts the frequency record for the selected word if the word is being tracked. The frequency is set by the individual user's word usage patterns. The invention does not track every word that is used. Some words do not have collisions and do not need a frequency count. The invention orders collisions that occur from equal frequency values relative to each other. The approach typically reduces the amount of memory required to store the frequency data to 1/7 of what is typically needed.

The Display Manager 301 must limit the frequencies for the tracked words because, otherwise, the counts might exceed the storage capacity of a register on the device. The system must adjust to a user's change in usage. The invention's aging algorithm goes through the recorded frequencies and discounts older usage frequencies. Older usages do not reflect the user's current habits and therefore count for less.

This section details the processes of learning the user's usage patterns and generating the displayed selection list that will resemble the user usage patterns.

Referring to Fig. 4, a preferred embodiment of the invention's database implementation contains a Linguistic Database (LDB) 401, a Manufacturer Database (MDB) 402, and a User Database (UDB) 403. The Linguistic Database 401 is a fixed pre-compiled database containing words ordered by their frequency of use as measured by a linguistic model.

The Manufacturer Database 402 is a custom database provided by the OEM. The Manufacturer Database 402 is optional and is provided when an OEM has needs for an additional custom database beyond what is supplied in the Linguistic Database 401.

The User Database 403 contains user defined words and a reordering database (RDB) which is a region of the User Database 403 that tracks the words that the user selects and their frequencies. Fig. 5 shows an exemplary depiction of some of the information 501 contained in the User Database 403 to track user word usage. The information lists all of the letters of each word 502, the frequency of use for the word 503, the input key sequence for the word 504, and the object number in the LDB 505 for words that are contained in the Linguistic Database 401.

The following terms are used in the text below:

UDB Reorder Word - An LDB word that was added to the RDB via word selection.

Active Word - A UDB Reorder Word that has an LDB field matching the current LDB. The concept of an active word is important when discussing the aging and garbage collection algorithms below.

Last Deletion Cut-Off Frequency - This is the frequency of the last UDB Reorder word deleted from the RDB by the garbage collection algorithm.

Reorder Words - Reorder Words are words whose frequencies are greater than or equal to the last deletion cut-off-frequencies. These words will be first on the selection display list and they include both UDB Added and UDB Reorder words.

UDB Added Words - UDB Added Word and UDB word have the same meaning; they refer to a word that has been added to the UDB by the user, either explicitly or b y the system detecting a novel word constructed by the user.

Reordering Database (RDB) - The portion of the UDB where UDB Reorder Words are stored. The RDB resides within the same memory or storage space of the UDB.

First-Word and Non First-Word - Both first-word and non first-word are LDB words. Relative to the LDB, the first-word is the first displayed LDB word and the non first-word is not.

The invention's Database Manager 301 learns a user's usage patterns by keeping track of the frequencies or usage counts of each of the UDB Added and UDB Reorder words. The usage counts (frequencies) of each word are updated and set according to the word's usage patterns. The usage counts are used to determine the relative position of a word in the displayed selection list, for garbage collection, and for aging.

The first step of keeping track of the usage pattern of an LDB word is by adding it to the RDB. This adding process is kicked in when a word is accepted. The key events for accepting a word are described below. Once an LDB word is added to the RDB it is called a UDB Reorder word. The rules for adding words to the RDB are described in Table 1 where the columns refer to the type of the first object in the selection list and the rows refer to the currently selected object type. The numbers in the table refers to the adding rules, below.

Table 1 – Adding Rules

	UDB Add Word	LDB Word	MDB Word
UDB Add Word	Rule_0	Rule_1	Rule_0
LDB Word	Rule_3	Rule_2	Rule_3
MDB Word	Rule_0	Rule_0	Rule_0

Based on the current selected object type and the object type of the first item in the displayed selection list, the table indicates the following rules:

- Rule_0 This is the case where the first object in the list is either a UDB Add word or a MDB word and the selected object is either the UDB Add word or a MDB word. In this case, neither the selected object nor the first object in the displayed selection list will be added to the RDB.

- Rule_1 In this case, the first object in the selection list is a LDB first-word and the selected object is a UDB Add word, the invention adds the LDB first-word (the first object in the displayed selection list) to the RDB.

- Rule_2 This applies when the first object in the displayed selection list is a LDB first-word and the selected object is a LDB non-first word. Both words are added to the RDB.

- Rule_3 When the selected object is a LDB word (either a LDB first or non-first word) and the first object in the displayed selection list is either a UDB Add or MDB word, then the selected LDB word is added to the RDB.

When an active word is accepted, the word is either added or not added to the RDB based on the adding rules. If the word is already in the RDB/UDB its frequency is bumped. The events for accepting a word are described as follows.

- When the context for building a word list is changed by either switching to a new language, by registering or un-registering the MDB, or by terminating the Database Manager.
- When a right arrow key is pressed.
- When a space key is pressed. Subsequent pressing of the space key right after a space or the arrow key is pressed does not have any affect since the word was already accepted.
- When on the user enters punctuation or explicit characters which don't match a known word.

A UDB Add word is added to the UDB when it is accepted as described above. The initial frequency is set to three use counts higher (this can be set to any value that sets it apart from the first-word) than the first-word. The UDB Added word frequency is bumped when the word is used.

The invention uses the "non-aggressive" learning principle, where a single usage of a non-first word should not beat the first-word to the first position in the displayed selection

list, by employing a gradual learning of the user's usage patterns to promote word ordering. The "non-aggressive" principle can be achieved by carefully updating and setting the frequencies.

- When adding a first-word and a non-first word to the RDB, the frequencies for the first-word is set so that it would take three (or a predetermined number) un-countered uses of the non first-word for the first-word to lose its first place position in the displayed selection list. The first time the user presses a space key to select a non-first word it is considered one use. For example, if both words were just added, then two more uses of the non first-word will move it to the first place in the displayed selection list.

- All non-first words start with the same initial frequencies. Their frequencies will b e increased or decreased by how often they are used. Frequencies are decreased during the Aging process.

- If selecting a first-word from the displayed selection list and it is already in the RDB, its frequency will be bumped.

- If attempting to add a first-word to the RDB (when selecting a non-first word) and the first-word is already in the RDB, its frequency will not increase. It is assumed that it is already in the correct position.

- The initial frequencies of a UDB Add Word are two use counts higher than the first-word. This delta value can be adjusted for different implementations.

At some time intervals, all of the UDB Add and UDB Reorder words will be aged. Aging means reducing their frequencies by some factor. How often the aging is performed is based on the unit of time interval used, the unit of time interval is maintained by the Database Manager. Since there is no concept of time in the database, the invention heuristically computes the time intervals by maintaining an update count. This update count is incremented by one every time a space key is pressed to select a word. When the update count has reached a certain value, the Aging process is kicked in. This is performed as follows:

- When the update count reaches its maximum (1000 in this case), the aging process kicks-in. Statistically, 1000 applies to a fast T9® user that can type 20 wpm using T9®, entering text with a sustained effort for 50 minutes. It is also about 50 messages of 20 words (~ 120 characters).

- The frequencies of all words are reduced by a (31/32) factor for aging. One skilled in the art will readily appreciate that any aging factor can be used to achieve the desired decay rate. For example, if the frequency of the non-first word was 54, then it will be reduced to 52 (54 x (31/32)).

For garbage collection, the "easy come, easy go" principle is used to first delete UDB Reorder Words and then UDB Add Words from the RDB and UDB storage space. Reorder words are preferred for deletion by a factor of 2. Thus, before a UDB Add word with X frequency is deleted, the invention first removes all Reorder words with frequencies less than or equal to 2*X. The garbage collection algorithm is described as follows:

1. Remove all UDB Add Words that have been marked as deleted.

2. If the amount of space free space after step 1 is greater than or equal to 1/8 of the total UDB data space, then garbage collection is done.

3. Otherwise, increment the Last Deletion Cut-Off Frequency and delete all UDB Reorder Words with this frequency and all UDB Add Words with half this frequency. This process continues until 1/8 of UDB space is free.

One skilled in the art will readily appreciate that the threshold value of 1/8 can be adjusted to reach a desired amount of UDB free space.

The new displayed selection list is composed of first (visually from top) Reorder Words, UDB, LDB, and MDB words (depending on the MDB fence). The MDB fence is used to set the maximum number of LDB words that can appear before the list of MDB words. This is to ensure that the OEM will have its words displayed. The number of Reorder Words in the displayed selection list is determined by the last deletion cut-off frequency or the non first-word initial frequencies, which ever is smaller - call this number RDB Count. All UDB Add and UDB Reorder Words whose frequencies are greater than or equal to RDB Count are Reorder Words and will appear first in the displayed selection list. The order of their appearance in the displayed selection list is hierarchically described as follows:

1. All Reorder Words whose frequencies are above the cut-off frequencies.
2. UDB terminal words.

3. LDB words up to the MDB fence.
4. MDB terminal words.
5. UDB stems.
6. MDB stems.

Tie Breaker refers to Reorder Words with the same frequencies when competing for the first 5 (number of sorted items) positions in the displayed selection list. The following rules are applied to break the tie:

1. If two UDB Add Words are tied, the last word added to the UDB wins.

2. If UDB Add and UDB Reorder Words are tied, the UDB Add Word wins.

3. If two UDB Reorder Words are tied, the word with the smaller LDB object number wins.

Each UDB Reorder Word is stored as a key sequence along with its LDB object number. With that information and the knowledge of which LDB they came from, the word can be re-constructed. This technique uses less memory to store a RDB word. Using only one nibble for a character rather one or two bytes per character.

One UDB Reorder Word should cost on average of eight bytes - two bytes frequency, one byte length, one byte LDB object number, one byte Language ID, plus three bytes for six characters word [average word length]. 4Kbytes of RDB space is capable of holding around 512 UDB Reorder Words, 3Kbytes would capable of holding around 384 words, 2Kbytes would hold around 256 words, and 1Kbytes holds around 128 words.

The user can to turn on and off the RDB. The behaviors are described as follows:

Turn on the RDB feature if it is not already turned on.

Turning off the RDB will have two effects:

- The displayed selection list falls to its original ordering.
- Any UDB Reorder words in the RDB will remain there until garbage collection.

One skilled in the art will readily appreciate that, although the term word has been used throughout the text above, the invention will equally apply to other linguistic input units such as a syllable or a phrase. For example, single syllables are input in Chinese and whole phrases are input in Japanese.

Although the invention is described herein with reference to the preferred embodiment, one skilled in the art will readily appreciate that other applications may be substituted for those set forth herein without departing from the spirit and scope of the present invention. Accordingly, the invention should only be limited by the Claims included below.